CLAIMS

We Claim:

1. A pipeline apparatus for use in a pulse output block that provides smooth, controlled transitions from wave-form to wave-form, wherein said pulse output block operates on the basis of a cycle time, a delta cycle time and a pulse count/width value, said apparatus comprising:

a first pipeline register for providing said cycle time;

a second pipeline register for providing said delta cycle time; and

a third pipeline register for providing said pulse count/pulse width value,

wherein said first, second and third pipeline registers are arranged to provide said cycle time, delta cycle time pulse count/width value, respectively, such that said pulse output block provides smooth, controlled transitions from wave-form to wave-form.

2. The apparatus according to claim 1, further comprising:

a clock generator for supplying a master time base clock;

a cycle time counter that is incremented on the rising edge of the time base clock.; and

a comparator for comparing a count of said counter with a predetermined cycle value, a result of a comparison by said comparator controls a period of an output waveform wherein, when said counter reaches said predetermined value then a current cycle is complete and a cycle done event is generated that indicates that a cycle is done.

3. The apparatus according to claim 2, further comprising an adder for adding a twos-complement value that is added to said predetermined cycle value at each cycle done event.

4. The apparatus according to claim 2, further comprising a pulse counter driven on each cycle done event to count the number of cycles generated on an output waveform of the pulse output block.

5. The apparatus according to claim 4, further comprising another comparator for comparing said pulse count /width value with a predetermined pulse count value to determine completion of a current pulse train.

6. The apparatus according to claim 2, wherein a duty-cycle of an output waveform of the pulse output block is controlled in response to an output of said cycle time counter, wherein, while a counter value of said cycle time counter is less than said pulse count/width value, said output waveform is biased is high and, while said counter value of said cycle time counter reaches or exceeds said output waveform, said output waveform is biased low.

7. The apparatus according to claim 1, wherein said apparatus smoothly transitions said pulse output block from a pulse train output waveform to a pulse width modulated waveform.

8. The apparatus according to claim 1, wherein said apparatus does not require that said pulse output block be stopped by software and then restarted.

9. The apparatus according to claim 1, further comprising a controller for changing said waveform of said pulse output block to provide ability to automatically ramp a value of the cycle time at a specified rate.

10. The apparatus according to claim 9, wherein said controller changes said waveform to allow for ramping a stepper motor.

11. A method for providing smooth, controlled transitions from wave-form to wave-form in a pulse output block that operates on the basis of a cycle time, a delta cycle time and a pulse count/width value, said method comprising the steps of:

pipelining said cycle time, delta cycle time and said pulse count/width value to said pulse output block; and

controlling said pipelining to control transitions from wave-form to waveform output from said pulse output block such that said transistions are smooth.

12. The method according to claim 11, further comprising the steps of:

counting a rising edge of a time base clock; and

controlling a period of an output waveform of said pulse block output circuit comparing a count of said counting with a predetermined value wherein, when said counter reaches said predetermined value, a current cycle of said output waveform is complete.

13. The method according to claim 12, further comprising the step of counting pulses driven on each cycle done event to count the number of cycles generated on an output waveform of the pulse output block.

14. The method according to claim 13, further comprising the step of comparing said pulse count /width value with a predetermined pulse value to determine completion of a current pulse train.

15. The method according to claim 12, wherein a duty cycle of an output waveform of the pulse output block is controlled in response to an output of said cycle time counter, further comprising the steps of biasing said output waveform high when a counter value of said cycle time counter is less than said pulse count/width value and biasing said output waveform low when said counter value of said cycle time counter reaches or exceeds said output waveform.

16. The method according to claim 11, further comprising the step of smoothly transitioning said pulse output block from a pulse train output waveform to a pulse width modulated waveform.

17. The method according to claim 11, further comprising the step of changing said waveform of said pulse output block to provide ability to automatically ramp a value of the cycle time at a specified rate.

18. The method according to claim 17, wherein said step of changing changes said waveform to ramp a stepper motor.

19. The method according to claim 11, further comprising the step of operating said pulse output block in a pulse train output mode for outputting pulse train waveforms.

20. The method according to claim 11, further comprising the step of operating said pulse output block in a pulse width modulated mode for outputting pulse width modulated waveforms.

21. An apparatus that provides a free port mode that allows a user to control a free port that links to a programmable logic controller, said apparatus comprising:

a port for linking to said programmable logic controller; and

a mode selectable by said user that allows said user to control said port, wherein said mode allows said user to configure a protocol for communication to said programmable logic controller through said port.

22. The apparatus according to claim 21, wherein said mode allows said user to configure a character based protocol.

23. The apparatus according to claim 21, whrein said mode allows said user to configure a number of bits per character.

24. The apparatus according to claim 21, whrein said mode allows said user to configure a parity of said protocol.

25. The apparatus according to claim 21, whrein said mode allows said user to configure a baud rate of said protocol.

26. The apparatus according to claim 21, wherein said programmable logic controller is a microcontroller.

27. The apparatus according to claim 21, further comprising a built-in protocol that is provided to the user for free port mode.

28. The apparatus according to claim 21, further comprising a predetermined receive instruction that defines a message content of data communicated thorugh said port.

29. A method for providing a free port mode that allows a user to control a free port that links to a programmable logic controller, said method comprising the steps of:

selecting a mode that allows said user to control said free port, wherein said mode allows said user to configure a protocol for

communication to said programmable logic controller through said free port; and

configuring said protocol for communicating to said programmable logic controller through said free port.

30. The method of claim 29, wherein said step of selecting implements an interrupt that interrupts said programmable logic controller and passes control of said free port to said user.

31. The method of claim 29, wherein said step of selecting passes control of said free port to a user program that instructs said programmable logic controller to function in a manner consistent with instructions of said user program.

32. The method of claim 29, wherein said programmable logic controller is a master that controls slave devices, wherein said step of selecting passes a token to said ~~master~~ to autorize said master to control said free port.

33. The method of claim 29, further comprising the step of setting a UART in accordance with predetermined settings for user control of said free port.

34. The method of claim 29, wherein said step of configuring provides a built-in protocol to the user for free port mode.

35. The method according to claim 29, wherein said step of configuring provides a predetermined receive instruction that defines a message content of data communicated thorugh said free port.

36. The method according to claim 29, further comprising the step of automatically returning to a normal mode after said free port mode is

complete wherein said programmable logic controller resumes control over said free port.

37. An apparatus for an n-bit protocol, n being an integer, for a modem wherein said n-bit protocol eliminates a parity bit of an n+1 bit protocol that includes said parity bit employed for parity checking an integrity of data transmitted in said n+1 bit protocol, said n-bit protocol comprising:

a data bit field of n-bits including at least one bit wherein said data bit field allocates each field to indicate a type of data for each of said n-bits thereby defining said n-bit protocol; and

wherein said n-bit protocol eliminates said parity bit of said n+1 bit protocol that includes said parity bit employed for parity checking an integrity of data transmitted in said n+1 bit protocol.

38. The apparatus according to claim 37, wherein said data bit field is a modified PPI protocol that is modified to exclude said parity bit.

39. The apparatus according to claim 37, wherein a plurality of blocks of n number of bits are transmitted according to said n-bit protocol, further comprising a check code implemented as at least one block of n number of bits to validate an integrity of at least one block of n number of bits.

40. The apparatus according to claim 39, wherein said check code is a CRC check code.

41. The apparatus according to claim 37, further comprising a connector for communicating said n-bit protocol with one pin grounded to signal use of said n-bit protocol.

42. The apparatus according to claim 41, further comprising a pull-up circuit connected to said pin for pulling up a potential of said pin to indicate other protocols.

43. The apparatus according to claim 37, wherein said n-bit protocol is a 10-bit protocol.

44. The apparatus accoding to claim 33, wherein said n-bit protocol includes 1 start bit, 8 data bits and 1 stop bit.

45. The apparatus according to claim 37, wherein said n-bit protocol is full duplex.

46. The apparatus according to claim 37, further comprising a transmission line through which data is transmitted in accordance with said n-bit protocol.

47. A method for generating an n-bit protocol, n being an integer, for a modem wherein said n-bit protocol eliminates a parity bit of an n+1 bit protocol that includes said parity bit employed for parity checking an integrity of data transmitted in said n+1 bit protocol, said n-bit protocol comprising:

allocating a data bit field of n-bits including at least one bit wherein said data bit field such that each field indicates a type of data for each of said n-bits thereby defining said n-bit protocol; and

eliminating said parity bit of said n+1 bit protocol that includes said parity bit employed for parity checking an integrity of data transmitted in said n+1 bit protocol.

48. The method according to claim 47, further comprising the step of converting an n+1 bit protocol to said n-bit protocol by truncating said parity bit of said n+1 bit protocol.

49. The method according to claim 47, further comprising the step of chekcing an integrity of data transmitted in accordance with said n-bit protocol by waiting a predetermined amount of time to elapse after receiving said data.

50. The method according to claim 47, wherein a plurality of blocks of n number of bits are transmitted according to said n-bit protocol, further comprising the step of generating a check code from at least one block of n number of bits whose contents indicate a validity of at least one block of n number of bits.

51. The method according to claim 50, wherein said step of generating generates a CRC check code.

Sub B1

52. A method for debugging a program in real time and while said program is executed by a programmable logic controller, said method comprising the steps of:

displaying a section of said program indicated by a user to be debugged;

saving original compiled code of said program;

compiling said section of said program to be debugged in another section of memory;

jumping to said another section of said memory during execution of said program when an instruction indicated to be debugged is to be executed; and

Sub B1

capturing a status of said instruction as it is executed, wherein said program is debugged in real time and while said program is executed by said programmable logic controller.

53. The method according to claim 52, further comprising the step of returning to said original compiled code of said program after said instruction indicated to be debugged is executed.

54. The method according to claim 53, further comprising the step of restoring said original compiled once said status is captured.

55. The method according to claim 52, further comprising the step of instrumenting each instruction compiled in said another section of memory.

56. The method according to claim 52, further comprising the step of storing a table relating instructions to relatively fastest boolean expressions, wherein said instructions are debugged relatively faster with said fastest boolean expressions.

57. The method according to claim 52, further comprising the step of providing a table of pointers to instructions of said original compiled code, wherein said instructions are located in memory relatively faster during debugging.

58. The method according to claim 52, further comprising the step of limiting a data size of each compiled instruction, wherein execution of said instructions to be debugged is relatively faster and memory required to store said instructions is minimized.

59. An apparatus that debugs a program in real time and while said program is executed by a programmable logic controller, said method comprising the steps of:

B1

an area of memory for saving original compiled code of said program;

another area of memory for storing a compiled section of said program to be debugged;

a branch that causes execution of said instruction to jump from said original compiled code to said another section of said memory during execution of said program when an instruction indicated to be debugged is to be executed; and

a circuit for capturing a status of said instruction as it is executed, wherein said program is debugged in real time and while said program is executed by said programmable logic controller.

60. The apparatus according to claim 59, further comprising a display for displaying said instructions to be debugged.

61. The apparatus according to claim 60, further comprising a table relating instructions to relatively fastest boolean expressions, wherein said instructions are debugged relatively faster with said fastest boolean expressions.

62. The apparatus according to claim 59, further comprising a table of pointers to instructions of said original compiled code, wherein said instructions are located in memory relatively faster during debugging.

63. An apparatus for high speed discrete output, said apparatus comprising:

a first optical transistor that, when turned on, connects a gate of an output transistor to a supply voltage such that the output transistor is turned OFF;

a second optical transistor that, when turned on, connects said gate of said output transistor to said supply, turning the output ON and driving the external load; and

logic that drives said first and second optical transistors as a complementary pair, so that for an output OFF state said first optical transistor is conducting and said second optical transistor is not, and in an output ON state said second optical transistor is conducting and said first optical transistor is not.

64. The apparatus according to claim 63, further comprising a relatively high impedance pull up that shut the output OFF in the case an that external load supply voltage is applied but internal power is not supplied.

65. The apparatus according to claim 64, further comprising a negative voltage regulator that provides a supply of gate drive voltage which is set to a value more negative than an external supply voltage.

66. The apparatus according to claim 65, wherien, on transitions between output OFF and output ON, said second optical transistor turns on relatively quickly, providing a path to discharge a capacitance of said gate and providing a low impedance path required for a quick turn off of said first optical transistor; and on an ON to OFF transition, said first optical transistor relatively quickly pulls the FET gate up and provides a low impedance path for quick turn off of said second optical transistor.

67. The apparatus according to claim 63, wherein said logic employs a digital logic gate for driving optical diodes of each optocoupler circuit through a plurality of parallel RC networks.

68. The apparatus according to claim 67, further comprising a resistor in each network that provides a limited DC current through the optical diode when a logic signal is in a proper state.

69. The apparatus according to claim 68, further comprising a capacitor in each RC network that allows for a pulse of current through an associated optical diode when a digital logic signal switches to said proper state to turn that diode on, thereby providing an initial relatively strong turn on of the respective optical transistor, followed by a steady state minimal optical drive.

70. The apparatus according to claim 63, further comprising a resistor / capacitor network that provides minimum load and decoupling of the negative voltage regulator.

71. The apparatus according to claim 63, further comprising capacitors across a collector / emitter of the each optical transistor that allows a path for noise coupled into the circuit via field wiring and the gate to by-pass the first and second optical transistors without causing undesired switching of those devices.

72. The apparatus according to claim 63, further comprising a zener diode coupled with a reverse blocking diode that provides inductive load clamping such that, as the gate attempts to turn off an inductive external load, inductive reaction will tend to drive the gate drain negative as the inductor attempts to maintain current.

73. The apparatus according to claim 71, wherein said zener diode allows this negative voltage to be transmitted to the gate circuit, wherein said gate is maintained ON in a linear region, allowing current to be maintained in the inductor without inducing a damaging voltage at the drain of the gate in order to dissipate a linear operating region voltage drop of the gate combined

with any resistance of the load stored inductor energy and turns off the circuit.

74. The apparatus according to claim 63, further comprising a blocking diode that protects the circuit against damage by erroneous reverse polarity connections of external load voltage supply.

75. A method for high speed discrete output, said method comprising the steps of:

driving first and second optical transistors as a complementary pair so that for an output OFF state said first optical transistor is conducting and said second optical transistor is not, and in an output ON state said second optical transistor is conducting and said first optical transistor is not.;

on transitions between output OFF and output ON, turning on said second optical transistor relatively quickly, thereby providing a path to discharge the a gate capacitance of a gate coupled to said complementary pair and providing the low impedance path required for a quick turn off of said first optical transistor; and

on an ON to OFF transition, causing said first optical transistor to quickly pulls said gate high and provides a low impedance path for a quick turn off of said second optical transistor.

76. The method according to claim 75, further comprising the step of driving the optical diodes of each optocoupler through parallel RC networks to provide a limited DC current through an optical diode of the respective optical tranistor when a logic signal is in a proper state.

77. The method according to claim 75, further comprising the step of sending a pulse of current through the associated optical diode when the digital logic signal switches to said proper state to turn that diode on.

78. The method according to claim 75, further comprising the step of initially turning on a respective optical transistor, followed by a steady state minimal optical drive.

79. The method according to claim 75, driving an initial relatively large conduction of gate charge and base charge of the opposite optical transistor, followed by a steady state conduction that is only sufficient to maintain the gate at a desired ON or OFF voltage level.

80. An apparatus for supporting customized function calls in a programmable logic controller having a predetermined set of function calls, said apparatus comprising:

a customized function call for directing said programmable logic controller; and

wherein, said customized function call is downloaded to said programmable logic controller and stored as part of said predetermined set of function calls.

81. The apparatus according to claim 80, further comprising a directory for reporting function calls of said programmable logic controller.

82. A method for supporting customized function calls in a programmable logic controller having a predetermined set of function calls, said method comprising the steps of:

creating a customized function call for directing said programmable logic controller;

downloading said customized function call to said programmable logic controller; and

storing said customized function call as part of said predetermined set of function calls.

83. The method according to claim 82, further comprising the step of reporting a directory of function calls of said programmable logic controller.